The Stata Journal (2021)
21, Number 2, pp. 539–554  DOI: 10.1177/1536867X211025838

Speaking Stata: Front-and-back plots to ease spaghetti and paella problems

Nicholas J. Cox
Department of Geography
Durham University
Durham, UK
n.j.cox@durham.ac.uk

Abstract. The spaghetti problem arises in graphics when multiple time series or other functional traces show mostly a tangled mess. The related paella problem (often experienced but not usually named as such) arises for multiple patterns combined in scatterplots. This column is a sequel to those in *Stata Journal* 10: 670–681 (2010) and 19: 989–1008 (2019). The focus is on what are here called front-and-back plots, in which each subset of data is shown separately with the other subsets as backdrop. The strategy is thus a hybrid of two more common strategies, showing each subset separately (juxtaposing) and showing subsets together (superimposing). A new command, `fabplot`, is introduced and used in examples.

Keywords: gr0087, fabplot, graphics, front-and-back plots, juxtaposing, superimposing, line plots, scatterplots, panel data, longitudinal data, quantile plots

1 Spaghetti and paella problems in statistical graphics

The spaghetti problem is easy to explain. Spaghetti plots are those showing many tangled lines—say, for multiple time series or other functional traces—which can be hard to distinguish and interpret. We may see broad collective patterns, but can we easily focus on individual series, too, or tell apart fine structure and mere noise?

This column is a sequel to a recent discussion of the spaghetti problem (Cox 2019). As promised then, it is also an update to discussion of a particular strategy discussed by Cox (2010).

The term "spaghetti" is often used informally in graphics discussions. Some token references to use in academic and professional literature were given in the 2019 column. Readers curious about earlier uses may appreciate an extra reference mentioning spaghetti, Zelazny (1985, 2001). In each edition of his book on business presentations (the dates just given are those of first and fourth editions), Zelazny gives (pp. 39, 111) examples in which a series of particular interest A is plotted in turn paired with each other series B, C, D, and E. That device is similar in spirit to the strategy used here but will not be explored further in this column.

As in Cox (2019), a related problem might be called the paella problem. Paella in scatterplots means that multiple point patterns for many groups are sufficiently mixed

up that comparisons are made difficult. In paella itself, the mixture is a feature, but in graphics it can be a problem.

2 Front-and-back plots

The focus in this column is on what are here called front-and-back plots, in which each subset of data is shown separately and prominently (in front, as it were) with the other subsets as backdrop. The strategy is thus a hybrid of two more common strategies, showing each subset separately (juxtaposing) and showing subsets together (superimposing). A new command, `fabplot`, is introduced and used in examples.

The need for a name is twofold. First, a name for use in Stata. Easy implementation of this strategy in Stata requires, or at least benefits from, a dedicated command. No such command was given in Cox (2010); that column explained how to approach the problem in Stata from first principles and gave example code. The command `subsetplot` (Cox 2014) was posted on the Statistical Software Components archive. The command `fabplot` formally published here is considered better. Either way, a Stata command name must obey a limit of 32 characters, start with a letter or an underscore, and use only those characters together with numeric characters. It should not repeat an existing command name. Those rules do not often bite. It is harder to think up a name that is concise and even catchy for those who might care. A name that people can pronounce easily would be a bonus too. I can think of worse names than `subsetplot`. Leaving out vowels, for example, is a surefire way to produce something that might be mistaken for Klingon. The problem with the name `subsetplot` is that it is not precise enough: how does the command offer something different from what is already standard? Turn and turn about, `fabplot` is on first reading a cryptic name, but once it is expanded and explained as "front-and-back plot", it may prove memorable enough.

Second, a name for a novel kind of plot. Not every kind of plot within statistical graphics needs a distinct name; otherwise, we would be tripping over terminology interminably. Nevertheless, this particular strategy is insufficiently known yet also often reinvented or rediscovered. An evocative name would do no harm in establishing it as a standard idea.

3 A line plot example: Investment in the Grunfeld data

The first example in Cox (2019) used the Grunfeld dataset bundled with Stata. That works as well as any to show the point of this strategy and indeed its limitations too. The dataset can be read into Stata with

```
. webuse grunfeld
```

The dataset includes various measures for 10 companies, each measured for 20 years. There are no missing values. If an idea does not work well with the Grunfeld data, it is unlikely to work well for larger or more complicated datasets.

Such datasets are often called longitudinal or panel data, depending partly on your field. The latter term raises mild ambiguity: when we say "panel", do we mean a subset of the data or panel in a graph with several panels? The ambiguity does not often bite hard. Terms such as "facet" are available for the graphical meaning but as yet do not seem common in Stata circles.

We focus here on line plots of investment as a time series. The syntax of `fabplot` always includes a subcommand that is a `twoway` subcommand.

```
. label variable invest "investment (million USD, 1947 prices)"
. fabplot line invest year, by(company) ysc(log) yla(1 10 100 1000)
> xtitle("")
```

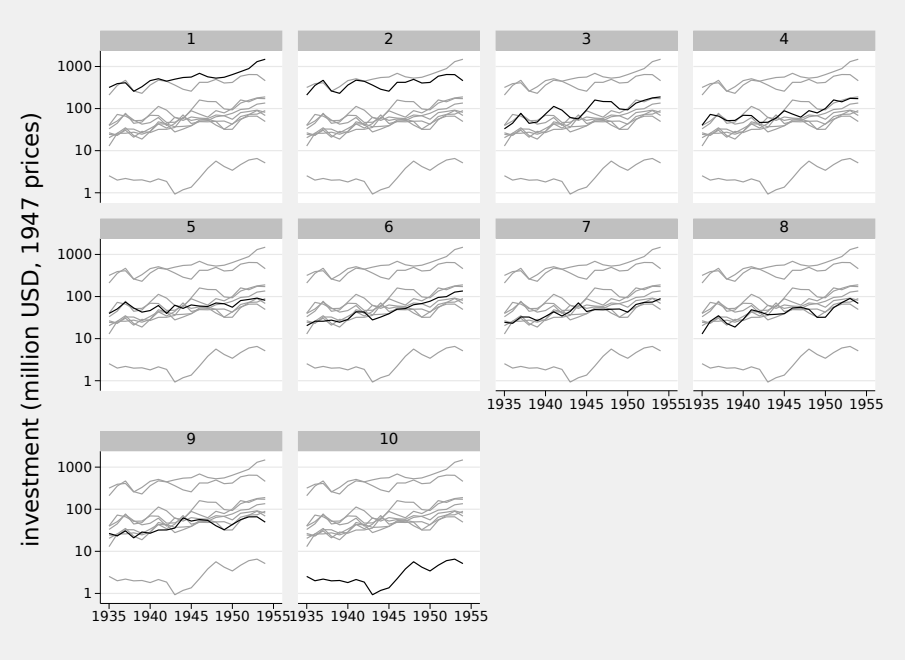

Figure 1. Investment time series for the Grunfeld panel data using a front-and-back plot

Figure 1 shows the result. The main idea is repeating each panel (here an individual company or firm) as a series in front but with all the other firms as backdrop. Company 1 is shown with companies 2 to 10 as backdrop, company 2 with companies 1 and 3 to 10 as backdrop, and so on.

As in Cox (2019), we added an informative variable label, adopted a logarithmic scale, and specified customized axis labels.

Without prompting, `twoway line` produces informative x-axis labels of 1935 1940 1945 1950 1955. Given that, the x-axis title—which, absent a variable label, would have been the variable name `year`—seems dispensable.

The defaults of `fabplot` are a little conservative. It may be a good idea to give more emphasis to each series in front or more contrast between front and back. The choice might depend on, say, how much space is available; whether color is allowed; or whether the graph remains visible to readers in a paper or is shown only briefly in a presentation. Color contrasts are not explored here beyond noting that the defaults for front and back with scatter, line, and connected plots are `blue` and `gs10`, respectively. Some advice on use of color was given in Cox (2019).

Varying the `twoway` subcommands used is an easy way to enhance contrast. In figure 2, the front series are, in Stata's terms, connected (meaning, shown by marker symbols as well as straight lines), while the back series remain shown as plain lines, as implied by the `line` subcommand.

```
. fabplot line invest year, by(company) ysc(log) yla(1 10 100 1000)
> xtitle("") front(connect)
```

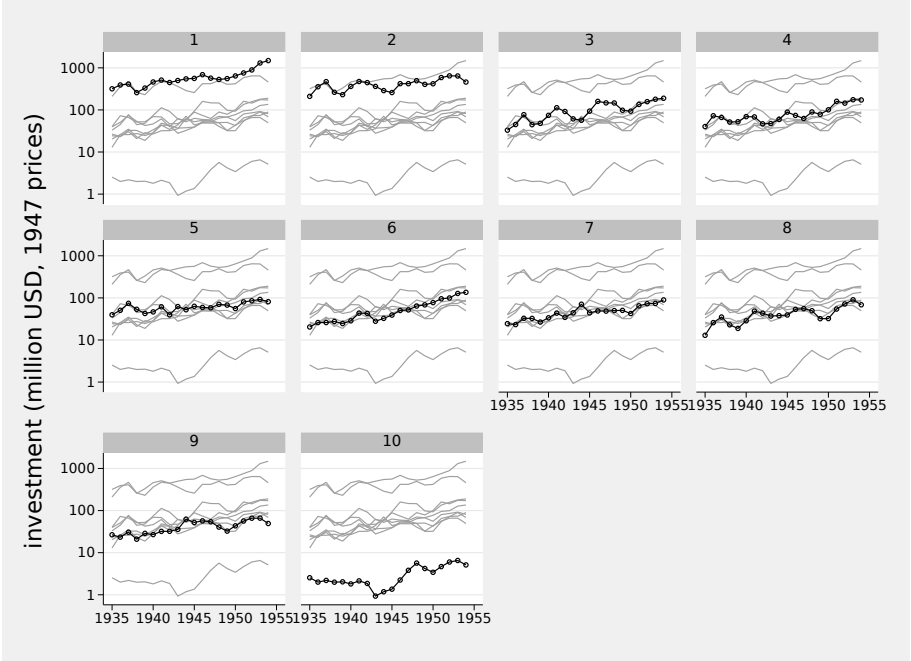

Figure 2. Investment time series for the Grunfeld panel data using a front-and-back plot. In this case, `connect` is used for the series in front, giving them more emphasis.

Another simple device is tuning line width. Figure 3 keeps the line choice but bumps up line width for the front series.

```
. fabplot line invest year, by(company) ysc(log) yla(1 10 100 1000)
> xtitle("") frontopts(lw(thick))
```

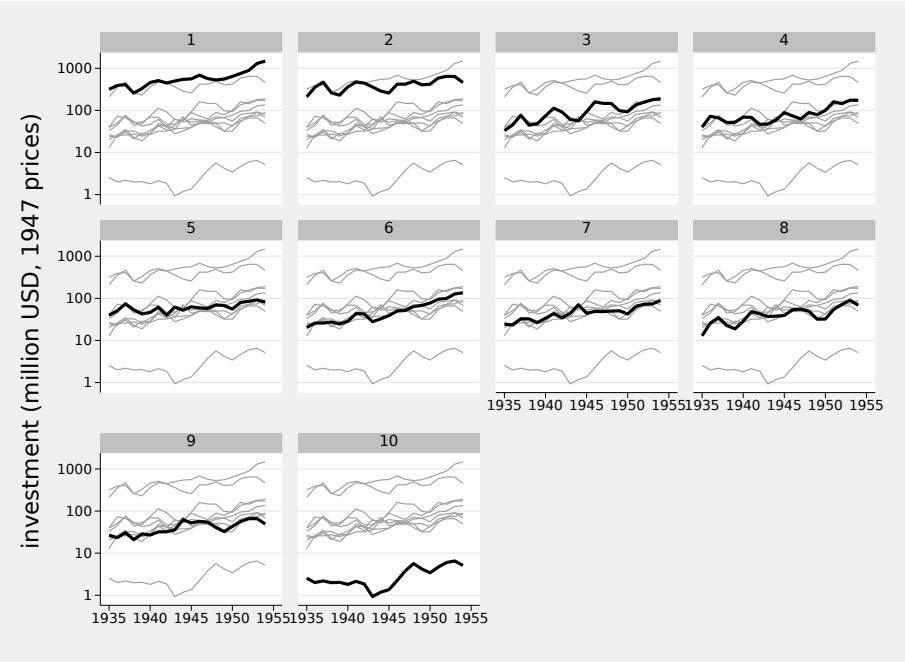

Figure 3. Investment time series for the Grunfeld panel data using a front-and-back plot. In this case, `lw(thick)` is used for the series in front, giving them more emphasis.

So far, the examples given have all matched the form

`fabplot` *twoway_subcommand yvar xvar* `...`

`fabplot` calls not matching that form are illegal, but not all legal possibilities will be helpful. Positively put, `fabplot` is written imagining use of one or more of `twoway line`, `twoway scatter`, and `twoway connected`.

4 Tradeoffs and the scope for selection

As in graphics generally, the tradeoff between subtle and strong contrasts can be delicate and difficult, balancing personal taste and the need for displays to be decoded easily and effectively. If the panels are all named, and the names mean something to researchers or readers, then one-to-one comparisons are likely to remain relevant, and tracking each series from one display to another remains desirable. That could be true of, say, countries

or other places in economic or environmental datasets. If the panels are anonymous, or their identifiers are of no interest, then other series may be no more than collective context, and being able to focus on details may be less important. That could be true of patients or anonymous subjects in medical or social datasets.

The number of panels being compared can pose difficulties. The Grunfeld dataset as a first example is large enough (10 companies) both to show the problem of easy and effective comparison and to show difficulties with any solution. Do readers want to scan 10 graphs, or can they be trusted to do so? What about 50 or 200 or 1000?

A device that can help is an ability to select panels. Suppose all panels are of some interest, but you are especially interested in only a few. `fabplot` supports `if` in standard Stata style, but it cannot meet this need. Concretely, imagine interest is focused on the four leading companies in the Grunfeld dataset. Specifying `if company <= 4` is allowed, but its result is to show company 1 with 2, 3, and 4 as backdrop; company 2 with 1, 3, 4 as backdrop; and so on. That could be desirable and needs no puff, but note that companies 5 to 10 never appear in the graph.

`fabplot` supports a `select()` option. The scope to use `if` and that option means that `fabplot` supports both kinds of selection. Thus, figure 4 shows graph panels for only 4 front series, but in each case with the other 9 series as backdrop.

```
. fabplot line invest year, by(company) ysc(log) yla(1 10 100 1000)
> xtitle("") frontopts(lw(thick)) select(company <= 4)
```

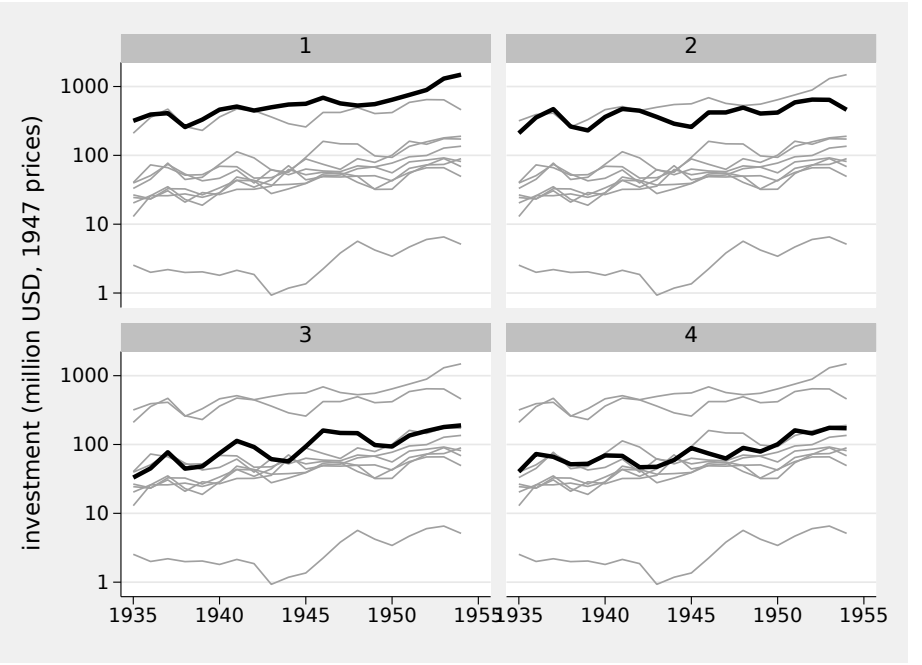

Figure 4. Investment time series for the Grunfeld panel data using a front-and-back plot. In this case, four leading companies are shown distinctly, with the other nine shown as backdrop.

The `select()` option gives a logical condition that must be true for a panel to be shown. As well as a condition specified on the fly such as `company <= 4`, it could be expressed using an indicator or dummy variable, already existing or constructed for the purpose, such as

```
. generate wanted = inlist(state, "CA", "FL", "NY", "TX")
```

for a dataset on the states of the United States in which California, Florida, New York, and Texas are to be shown distinctly—or

```
. generate wanted = inlist(country, "DE", "FR", "GB", "IT")
```

for a dataset on European countries in which Germany, France, Britain, and Italy are to be flagged.

It is a matter of taste only, but it seems to me that graphs with 2, 4, 6, and 9 panels can look quite good. I am even tempted sometimes to avoid graphs with 3, 5, 7, or 8 panels, usually positively by including an extra panel or two. Otherwise put, if space is available, you might as well use it to show some data.

5 A scatterplot example: auto.dta

Let us look at an example of front-and-back plotting applied to scatterplots. We switch to `auto.dta`. For this kind of dataset, using some kind of categorical variable to separate subsets is a standard strategy (for example, Cox [2005]). This can work well, especially if subsets segregate easily, as in the famous iris dataset, or the categorical variable is an indicator variable with just two distinct values. In `auto.dta`, the pattern for such scatterplots with the `foreign` variable is usually simple enough to think about. That variable distinguishes cars that are foreign (made outside the United States) and those that are domestic (made inside the United States). However, plots distinguishing categorical variables with many distinct values are often just a mess. Repair record has five nonmissing categories, so it provides our example.

```
. sysuse auto, clear
(1978 Automobile Data)
. fabplot scatter mpg weight, by(rep78)
```

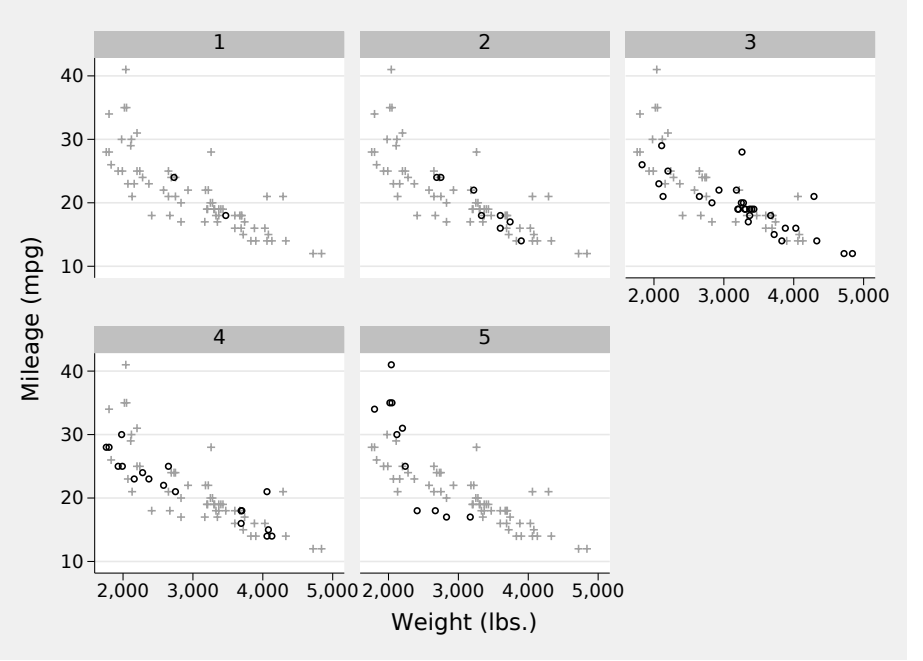

Figure 5. Scatterplot for miles per gallon and weight by repair record for cars in `auto.dta` shown as a front-and-back plot

Figure 5 is a basic front-and-back plot, but a little more work can make it more effective. The idea here is to suppress the marker symbols and replace them with marker labels at the same places and magnified a little. That improves comparison both of each subset in its own graph panel with its backdrop and of subsets across panels. Figure 6 is the result. Bare numbers 1 2 3 4 5 may seem prosaic, but they are

the data values in this case. In principle, any numeric or string variable may be used as a marker label, and text labels that are one, two, or three characters long can work well (Cox 2005, 2019). Fuller names may be in order so long as you have the space to show them without messy overlaps.

```
. fabplot scatter mpg weight, frontopts(ms(none) mla(rep78)
> mlabsize(medlarge) mlabpos(0)) by(rep78)
```

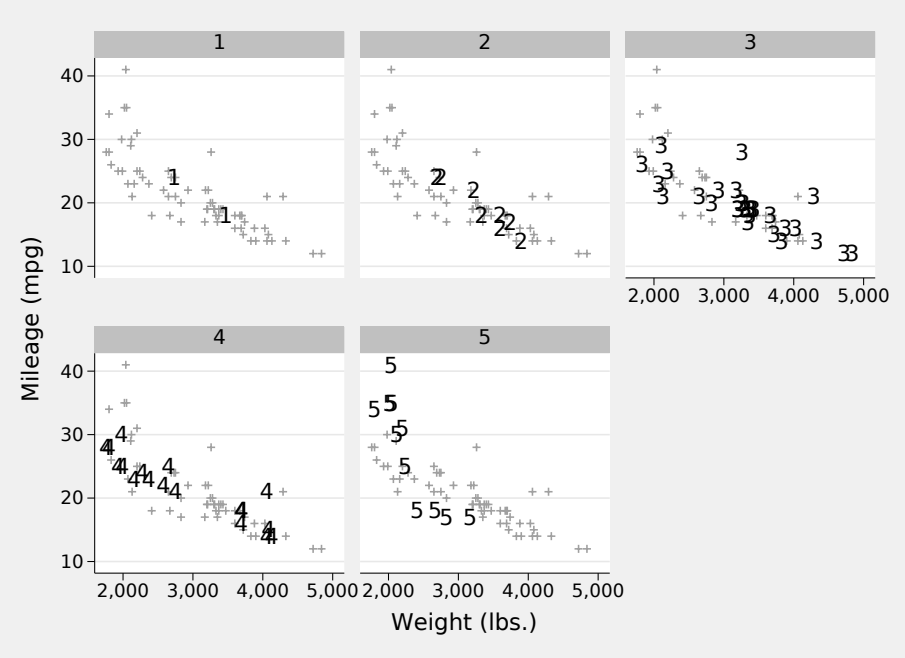

Figure 6. Scatterplot for miles per gallon and weight by repair record for cars in `auto.dta` shown as a front-and-back plot. Marker labels are used as self-explanatory symbols.

6 A more challenging example: Gumbel quantile plots

The major example in Cox (2010) consisted of a composite Gumbel quantile plot for six distributions of annual windspeed maximums from various stations in Texas, Alabama, Florida, and Georgia. As explained there with references or through Cox (2007) and its references, such a plot ensures that a sample perfectly matching a Gumbel distribution would plot as data points following a straight line. In this section, that example is revisited using `fabplot`.

`fabplot` is a framework for calling up a `twoway` command or commands with temporarily restructured data. The examples so far have all focused on using variables already in a dataset, using common or garden line and scatterplots. Many more specialized

plots can be obtained through other Stata commands, whether official or community contributed. In most other cases, the task reduces to first doing some calculations with the data and then calling up `twoway` to draw the graph. Commands for quantile plots fit this pattern exactly.

You may wish to skim or skip ahead to the next chunk of code if you are already familiar with the idea of quantile plots (still often called probability plots).

Quantile plots typically show on one axis raw data or occasionally those raw data on a transformed scale. That typically requires little or no work from a researcher. On the other axis is shown an estimate of the associated cumulative probability, the fraction of the data less than or equal to each ordered value, often called the plotting position in graphical contexts, or percentile rank in numerical reporting. That estimate requires more work if you are not using a preexisting command. Often, the estimate is shown on some transformed scale, typically as the quantile of a reference distribution, as in this example.

The small issues surrounding plotting positions are illuminated by imagining a toy sample of seven observations with no ties, whose values would be ranked 1 to 7. Let us agree that the median for such an example dataset must have rank 4 and should have associated cumulative probability or plotting position 0.5. Possible rules for plotting positions rank / sample size or (rank $-$ 1) / sample size both fail because they would yield plotting positions for the median of 4/7 or 3/7, respectively. The oldest solution for this tiny dilemma, which goes back at least to Galton (1883), is to split the difference and use (rank $-$ 0.5) / sample size. Other solutions can be suggested on other grounds that give the same plotting position for the median of an odd number of values and also treat the upper and lower halves of the data symmetrically. A more subtle issue arises: plotting positions should not ever be 0 or 1, because only if probability distributions have finite bounds will the corresponding quantiles, indicating the minimum and maximum possible values, be determinate. Plotting positions strictly within $(0, 1)$ leave scope for lower or higher values than those observed in a sample.

There is a continuing and even agitated literature discussing the merits and demerits of various plotting position rules. For some examples and more references, see Cox (2016).

A pragmatic position, which would not satisfy all of those who have written on this issue, starts with insistence that most uses of quantile plots are descriptive or exploratory. If the appearance or interpretation of such plots depends sensitively on choice of plotting position rule, then your sample is too small or too awkward to indicate very much reliably. The discussion is not trivial, however, if the issue is reliable estimation, usually extrapolation, of extreme quantiles.

Stata lacks an official command or function for plotting position calculation. That is not much of a limitation. It is even a feature insofar as it allows and indeed encourages researchers to think through what they want and make their choice explicit in code.

In real datasets, ties and missing values are entirely possible, so it is prudent to use `egen` functions that act sensibly if such exist. A further strong advantage of `egen`

here is canned support for groupwise calculations. Just to show some caprice, we do not emulate the code in Cox (2010) exactly but use Galton's rule for plotting positions. Once again, see Cox (2007, 2016) if you seek more discussion or if the rationale of the code is unclear.

```
. use http://www.stata-journal.com/software/sj10-4/gr0046/windspeed.dta, clear
(Hosking, J.R.M. and Wallis, J.R. 1997. Regional frequency analysis. C.U.P. p.31)
. egen rank = rank(windspeed), by(place) unique
. egen count = count(windspeed), by(place)
. generate pp = (rank - 0.5)/ count
. label variable pp "fraction of data"
. generate gumbel = -ln(-ln(pp))
. label variable gumbel "Gumbel reduced variate"
. fabplot scatter windspeed gumbel, by(place)
```

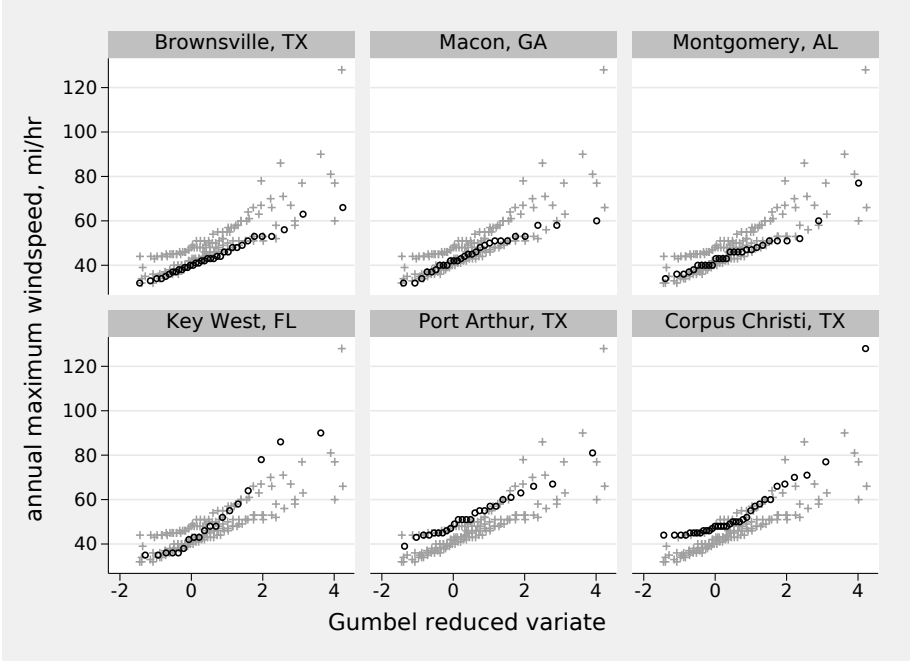

Figure 7. Quantile plots for annual windspeed maximums for six stations shown as a front-and-back plot

Figure 7 is the result. The main point for current purposes is simply that `fabplot` has made the coding problem simpler. Naturally, if you liked the idea, you could wire similar code into a do-file or program of your own.

7 Roots and relations

I gather together here a bundle of references to applications of this idea. Most of these cite no other applications, so a fair surmise is that it has been repeatedly rediscovered or reinvented and hence that yet other applications may abound.

Front-and-back plots are a special case of what Tufte (1990, 1997, 2001) called small multiples, in which a display contains several panels, each a variation on a theme.

The longest root I have encountered of front-and-back plots belongs to a related idea in dynamic graphics dubbed "alternagraphics" by Tukey (1973). See also Tukey (1983), Tukey and Tukey (1985), or Monmonier (1993). Perhaps more accessible to many readers is a review by Becker, Cleveland, and Wilks (1987). They explain the principle (pp. 357–358): "At a given moment in time the viewer can identify some of the subsets and the selection of identified subsets can be changed quickly. There are many ways of implementing this idea. One is to cycle through the subsets showing each for a short time period .... Another technique is to show all of the data at all times and have the cycling consist of a highlighting of one subset at each stage. A third technique is to provide the data analyst with the capability of turning any subset on or off ...."

Cleveland (1985, 74, 203, 205, 268) shows graphs in which summary curves for groups are repeated with data shown separately for each group. (Note: these graphs do not appear in Cleveland [1994].) The same idea is used in Wallgren et al. (1996, 47, 69).

Other examples can be found in Koenker (2005, 12–13); Carr and Pickle (2010, 85); Yau (2013, 224); Rougier, Droettboom, and Bourne (2014); Schwabish (2014; 2017, 98); Knaflic (2015, 233); Unwin (2015, 121, 217); Berinato (2016, 74); Cairo (2016, 211); Camões (2016, 354); Standage (2016, 177); Wickham (2016, 157); Kriebel and Murray (2018, 303); Grant (2019, 52); Koponen and Hildén (2019, 101); and Tufte (2020, 26).

Between submission of this column and final proofreading, I stumbled across examples in *The Guardian* of February 6, 2021 (https:// www.theguardian.com / business / 2021/feb/06/is-big-tech-now-just-too-big-to-stomach), and *The Economist* of April 10, 2021 (https: // www.economist.com / graphic-detail / 2021 / 04 / 10 / our-house-price-forecast-expects-the-global-rally-to-lose-steam). Researchers should be taking note when good journalism raises standards in statistical visualization.

Readers knowing interesting or useful examples or discussions, especially early in date or comprehensive in detail, are welcome to email the author.

8 Syntax for fabplot

`fabplot` *subcommand yvar xvar* [*if*] [*in*] , `by(`*byvar* [, *byopts*]`)`
[`select(`*condition*`)` `front(`*twoway_command*`)` `frontopts(`*twoway_options*`)`
graph_options]

`fabplot` produces an array of `scatter` or other `twoway` plots for *yvar* versus *xvar* according to a further variable *byvar*. There is one plot for observations for each distinct subset of *byvar* in which data for that subset are highlighted (shown at the front or in the foreground, as it were) and the rest of the data are shown as backdrop. The name `fabplot` can thus be understood as indicating a plot showing some observations in each panel in the `f`ront or as `f`oreground and the others as `b`ackdrop or `b`ackground.

8.1 Options

`by(`*byvar* [`,` *byopts*]`)` specifies a numeric or string variable *byvar* defining the distinct subsets being plotted. `by()` is required. Options of `by()` may be specified in the usual way: see the help for *by_option*.

`select(`*condition*`)` specifies a true-or-false condition, such as one referring to *byvar*, selecting which panels are shown. This is best explained with a concrete example. You have 10 companies but wish to display only panels for the 4 most interesting or important, yet in each case data for the other 9 companies should be shown as backdrop. Note that a standard `if` qualifier cannot match this mix of choices.

`front(`*twoway_command*`)` specifies a `twoway` command used to plot observations in each distinct subset as front or foreground.

`frontopts(`*twoway_options*`)` specifies options of `twoway` tuning the front or foreground plot of each distinct subset.

graph_options are options of `twoway` used to display observations for the rest of the data in each plot.

9 Conclusion

Plotting data with a subset structure is a long-standing problem in statistical graphics. As one solution, front-and-back plots have long roots yet remain used and appreciated only occasionally. This column is further publicity for the idea, which is the main story, and for a distinct name and for a Stata implementation. The aim of the hybrid strategy is to get the best of both worlds—the clarity imparted by separate focus on each subset and the context provided by seeing that subset compared with all the other data.

10 Acknowledgments

Naomi B. Robbins helped with the Zelazny references from her personal copies. Antony Unwin underlined related tactics in dynamic graphics.

11 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 21-2
. net install gr0087     (to install program files, if available)
. net get gr0087         (to install ancillary files, if available)
```

12 References

Becker, R. A., W. S. Cleveland, and A. R. Wilks. 1987. Dynamic graphics for data analysis. *Statistical Science* 2: 355–395. Reprinted in *Dynamic Graphics for Statistics*, 1988, ed. Cleveland, W. S., and M. E. McGill, 1–72. Belmont, CA: Wadsworth. https://doi.org/10.1214/ss/1177013104.

Berinato, S. 2016. *Good Charts: The HBR Guide to Making Smarter, More Persuasive Data Visualizations*. Boston: Harvard Business Review Press.

Cairo, A. 2016. *The Truthful Art: Data, Charts, and Maps for Communication*. San Francisco: New Riders.

Camões, J. 2016. *Data at Work: Best Practices for Creating Effective Charts and Information Graphics in Microsoft Excel*. San Francisco: New Riders.

Carr, D. B., and L. W. Pickle. 2010. *Visualizing Data Patterns with Micromaps*. Boca Raton, FL: Chapman & Hall/CRC.

Cleveland, W. S. 1985. *The Elements of Graphing Data*. Monterey, CA: Wadsworth.

———. 1994. *The Elements of Graphing Data*. Rev. ed. Summit, NJ: Hobart.

Cox, N. J. 2005. Stata tip 27: Classifying data points on scatter plots. *Stata Journal* 5: 604–606. https://doi.org/10.1177/1536867X0500500412.

———. 2007. Stata tip 47: Quantile–quantile plots without programming. *Stata Journal* 7: 275–279. https://doi.org/10.1177/1536867X0700700213.

———. 2010. Speaking Stata: Graphing subsets. *Stata Journal* 10: 670–681. https://doi.org/10.1177/1536867X1101000408.

———. 2014. subsetplot: Stata module for plots for each subset with rest of the data as backdrop. Statistical Software Components S457919, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457919.html.

———. 2016. FAQ: How can I calculate percentile ranks? http://www.stata.com/support/faqs/statistics/percentile-ranks-and-plotting-positions/.

———. 2019. Speaking Stata: Some simple devices to ease the spaghetti problem. *Stata Journal* 19: 989–1008. https://doi.org/10.1177/1536867X19893641.

Galton, F. 1883. *Inquiries into Human Faculty and its Development*. London: Macmillan.

Grant, R. 2019. *Data Visualization: Charts, Maps, and Interactive Graphics*. Boca Raton, FL: CRC Press.

Hosking, J. R. M., and J. R. Wallis. 1997. *Regional Frequency Analysis: An Approach Based on L-Moments*. Cambridge: Cambridge University Press.

Knaflic, C. N. 2015. *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Hoboken, NJ: Wiley.

Koenker, R. 2005. *Quantile Regression*. New York: Cambridge University Press.

Koponen, J., and J. Hildén. 2019. *The Data Visualization Handbook*. Espoo: Aalto ARTS Books.

Kriebel, A., and E. Murray. 2018. *#MakeoverMonday: Improving How We Visualize and Analyze Data, One Chart at a Time*. Hoboken, NJ: Wiley.

Monmonier, M. 1993. Adapting the alternagraphics concept to dynamic cartography. *Journal of the Pennsylvania Academy of Science* 67: 16–20.

Rougier, N. P., M. Droettboom, and P. E. Bourne. 2014. Ten simple rules for better figures. *PLOS Computational Biology* 10: e1003833. https://doi.org/10.1371/journal.pcbi.1003833.

Schwabish, J. A. 2014. An economist's guide to visualizing data. *Journal of Economic Perspectives* 28: 209–234. https://doi.org/10.1257/jep.28.1.209.

———. 2017. *Better Presentations: A Guide for Scholars, Researchers, and Wonks*. New York: Columbia University Press.

Standage, T. 2016. *Go Figure: The Economist Explains: Things You Didn't Know You Didn't Know*. London: Profile Books.

Tufte, E. R. 1990. *Envisioning Information*. Cheshire, CT: Graphics Press.

———. 1997. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire, CT: Graphics Press.

———. 2001. *The Visual Display of Quantitative Information*. 2nd ed. Cheshire, CT: Graphics Press.

———. 2020. *Seeing with Fresh Eyes: Meaning, Space, Data, Truth*. Cheshire, CT: Graphics Press.

Tukey, J. W. 1973. Some thoughts on alternative graphic displays. Technical Report Series 2 No. 45, Department of Statistics, Princeton University.

———. 1983. Another look at the future. In *Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface*, ed. K. W. Heiner, R. S. Sacher, and J. W. Wilkinson, 2–8. New York: Springer. Reprinted in *The Collected Works of John W. Tukey. Volume V: Graphics: 1965–1985*, 1988, ed. W. S. Cleveland, 405–418. Pacific Grove, CA: Wadsworth and Brooks/Cole.

Tukey, J. W., and P. A. Tukey. 1985. Computer graphics and exploratory data analysis: An introduction. In *Proceedings of the Sixth Annual Conference and Exposition, Volume III, Technical Sessions*, 773–785. Fairfax, VA: National Computer Graphics Association. Reprinted in *The Collected Works of John W. Tukey. Volume V: Graphics: 1965–1985*, 1988, ed. W. S. Cleveland, 419–436. Pacific Grove, CA: Wadsworth and Brooks/Cole.

Unwin, A. 2015. *Graphical Data Analysis with R*. Boca Raton, FL: Taylor & Francis.

Wallgren, A., B. Wallgren, R. Persson, U. Jorner, and J.-A. Haaland. 1996. *Graphing Statistics and Data: Creating Better Charts*. Newbury Park, CA: SAGE.

Wickham, H. 2016. *ggplot2: Elegant Graphics for Data Analysis*. 2nd ed. Cham, Switzerland: Springer.

Yau, N. 2013. *Data Points: Visualization That Means Something*. Indianapolis, IN: Wiley.

Zelazny, G. 1985. *Say It with Charts: The Executive's Guide to Successful Presentations*. Homewood, IL: Dow Jones-Irwin.

———. 2001. *Say It with Charts: The Executive's Guide to Visual Communication*. 4th ed. New York: McGraw–Hill.

About the author

Nicholas Cox is a statistically minded geographer at Durham University. He contributes talks, postings, FAQs, and programs to the Stata user community. He has also coauthored 16 commands in official Stata. He was an author of several inserts in the *Stata Technical Bulletin* and is an editor of the *Stata Journal*. His "Speaking Stata" articles on graphics from 2004 to 2013 have been collected as *Speaking Stata Graphics* (2014, College Station, TX: Stata Press).